

Formal Abstraction of Linear Systems via Polyhedral Lyapunov Functions[★]

Xuchu Ding^{*} Mircea Lazar^{**} Calin Belta^{***}

^{*} *Dept. of Systems at United Technologies Research Center, East Hartford, CT 06108, USA. Email: dingx@utrc.utc.com*

^{**} *Dept. of Electrical Engineering at Eindhoven University of Technology, Eindhoven, The Netherlands. Email: m.lazar@tue.nl*

^{***} *Dept. of Mechanical Engineering at Boston University, Brookline, MA 02446, USA. Email: cbelta@bu.edu*

Abstract: In this paper we present an abstraction algorithm that produces a finite bisimulation quotient for an autonomous discrete-time linear system. We assume that the bisimulation quotient is required to preserve the observations over an arbitrary, finite number of polytopic subsets of the system state space. We generate the bisimulation quotient with the aid of a sequence of contractive polytopic sublevel sets obtained via a polyhedral Lyapunov function. The proposed algorithm guarantees that at iteration i , the bisimulation of the system within the i -th sublevel set of the Lyapunov function is completed. We then show how to use the obtained bisimulation quotient to verify the system with respect to arbitrary Linear Temporal Logic formulas over the observed regions.

1. INTRODUCTION

In recent years, there has been a trend to bridge the gap between control theory and formal methods. Control theory allows verifications of “simple” specifications (such as stability or reachability) for “complex” dynamical systems with a possibly infinite state space, while formal verification methods enable validation of a “simple” finite system in a “complex” (rich and expressive) specification language. Recent studies in the area of abstraction allow one to model the behaviors of complex dynamical systems as finite systems, so that formulas in a rich specification language such as Linear Temporal Logic (LTL) can be used to analyze, verify and control the behavior of the system, with applications in areas such as robotics [Belta et al., 2007], multi-agent control systems [Loizou and Kyrkiakopoulos, 2004] and bioinformatics [Batt et al., 2005].

In this paper, we focus on autonomous (without inputs) linear systems, and we aim to generate a finite bisimulation abstraction of the system within some relevant subset of the state space. Since the bisimulation quotient preserves the language of the original infinite state system, it can be readily used for system verification.

Our approach relies upon the existence of a *polyhedral* Lyapunov function, which is non-conservative for stable linear systems, and we take advantage of the recent method by Lazar [2010] to construct such Lyapunov functions. The polyhedral Lyapunov function is used to generate a sequence of sublevel sets, which are contractive polytopes. We propose to partition the state space with respect to these polytopic sublevel sets, as they allow us to incrementally generate the bisimulation quotient of the entire

relevant state space. As the abstraction algorithm iterates, we guarantee that the bisimulation quotient is generated for an increasing larger sublevel set, with no “holes” in the covered state space. The polytopic sublevel sets also ensure that the algorithm proposed in this paper only requires polytopic operations, and can be tractably implemented for systems in realistic and practical applications.

This work is related to relevant works on the construction of finite quotient for infinite systems, such as controlled linear systems [Tabuada and Pappas, 2006, Pappas, 2003] and hybrid systems [Alur et al., 2000]. The bisimulation problem in general does not terminate [Milner, 1989]. We side-step this issue by only considering the system behavior within a relevant state space, i.e., in between two positive invariant compact sets that contain the origin in their interior. Such positive invariant sets with arbitrary sizes can be immediately obtained from the polyhedral Lyapunov function as polytopic sublevel sets (*i.e.*, polytopes) centered at the origin. Therefore, the bisimulation algorithm can capture any relevant subset of the state space. This also directly gives a trade-off between the size of the bisimulation quotient and the size of the relevant state space being analyzed.

Another conceptually related work is Sloth and Wisniewski [2010], where two orthogonal (quadratic) Lyapunov functions were used for the abstraction of continuous-time Morse-Smale systems (including hyperbolic linear systems) to timed automata. Besides targeting general discrete-time linear systems, the main difference between Sloth and Wisniewski [2010] and the approach proposed in this paper comes from the usage of *polyhedral Lyapunov functions*. This turns out to be beneficial, as it removes the need for two orthogonal Lyapunov functions and it results in a tractable implementation.

[★] This work is partially supported by the ONR-MURI Award N00014-09-1051 at Boston University and the STW Veni Grant 10230 at Eindhoven University of Technology.

The rest of the paper is organized as follows. We introduce preliminaries in Sec. 2 and formulate the problem in Sec. 3. We present the algorithm to generate the bisimulation quotient in Sec. 4, and we show in Sec. 5 how the resulting bisimulation quotient can be used to verify the system behavior against formulas in LTL. Conclusions are summarized in Sec. 6.

2. PRELIMINARIES

For a set \mathcal{S} , $\text{int}(\mathcal{S})$, $\partial(\mathcal{S})$, $\text{Co}(\mathcal{S})$, $|\mathcal{S}|$, and $2^{\mathcal{S}}$ stand for its interior, boundary, convex hull, cardinality, and power set, respectively. For $\lambda \in \mathbb{R}$ and $\mathcal{S} \subseteq \mathbb{R}^n$, let $\lambda\mathcal{S} := \{\lambda x \mid x \in \mathcal{S}\}$. We use \mathbb{R} , \mathbb{R}_+ , \mathbb{Z} , and \mathbb{Z}_+ to denote the sets of real numbers, non-negative reals, integer numbers, and non-negative integers. For $m, n \in \mathbb{Z}_+$, we use \mathbb{R}^n and $\mathbb{R}^{m \times n}$ to denote the set of column vectors and matrices with n and $m \times n$ real entries. For a vector $x \in \mathbb{R}^n$, $[x]_i$ denotes the i -th element of x and $\|x\|_\infty = \max_{i=1, \dots, n} |[x]_i|$ denotes the infinity norm of x . For a matrix $Z \in \mathbb{R}^{l \times n}$, let $\|Z\|_\infty := \sup_{x \in \mathbb{R}^n \setminus 0} \frac{\|Zx\|_\infty}{\|x\|_\infty}$ denote its infinity norm.

A n -dimensional *polytope* \mathcal{P} (see, e.g., Ziegler [1995]) in \mathbb{R}^n can be described as the convex hull of $n+1$ affinely independent points in \mathbb{R}^n . Alternatively, \mathcal{P} can be described as the intersection of k , where $k \geq n+1$, closed half spaces, i.e., there exists $k \geq n+1$ and $H_{\mathcal{P}} \in \mathbb{R}^{k \times n}$, $h_{\mathcal{P}} \in \mathbb{R}^k$, such that

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid H_{\mathcal{P}}x \leq h_{\mathcal{P}}\}. \quad (1)$$

We assume polytopes in \mathbb{R}^n are n -dimensional unless noted otherwise. The set of boundaries of a polytope \mathcal{P} are called *facets*, denoted by $f(\mathcal{P})$, which are themselves $(n-1)$ -dimensional polytopes. A *semi-linear* set (sometimes called a *polyhedron* in literature) in \mathbb{R}^n is defined as finite unions, intersections and complements of sets $\{x \in \mathbb{R}^n \mid a^T x \sim b, \sim \in \{=, <\}\}$, for some $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Note that a convex and bounded semi-linear set is equivalent to a polytope with some of its facet removed.

2.1 Transition systems and bisimulations

Definition 2.1. A transition system (TS) is a tuple $\mathcal{T} = (Q, \rightarrow, \Pi, h)$, where

- Q is a (possibly infinite) set of states;
- $\rightarrow \subseteq Q \times Q$ is the set of transitions;
- Π is a finite set of observations; and
- $h : Q \rightarrow 2^\Pi$ is the observation map.

We denote $x \rightarrow x'$ if $(x, x') \in \rightarrow$. We assume \mathcal{T} to be non-blocking, i.e., for each $x \in Q$, there exists $x' \in Q$ such that $x \rightarrow x'$. A *trajectory* of a TS from an initial state x_0 is an infinite sequence $\mathbf{x} = x_0x_1\dots$ where $x_k \rightarrow x_{k+1}$ for all $k \in \mathbb{Z}_+$. A trajectory \mathbf{x} generates a word $\mathbf{o} = o_0o_1\dots$, where $o_k = h(x_k)$ for all $k \in \mathbb{Z}_+$.

The TS \mathcal{T} is *finite* if $|Q| < \infty$, otherwise \mathcal{T} is *infinite*. Moreover, \mathcal{T} is *deterministic* if for all $x \in Q$, there exists at most one $x' \in Q$ such that $x \rightarrow x'$, otherwise, \mathcal{T} is called *non-deterministic*. Given a set $X \subseteq Q$, we define:

$$\text{Pre}_{\mathcal{T}}(X) = \{x \in Q \mid \exists x' \in X, x \rightarrow x'\}, \quad (2)$$

i.e., $\text{Pre}_{\mathcal{T}}(X)$ is the subset of Q that reaches X in one step. At a state $x \in Q$, the set of all words generated by

trajectories originating from x is called the *language* of \mathcal{T} originating at x , which is denoted by $\mathcal{L}_{\mathcal{T}}(x)$. We also denote by $\mathcal{L}_{\mathcal{T}}(X)$ the language of \mathcal{T} originating from states in a subset $X \subseteq Q$.

States of a TS can be related by a relation $\sim \subseteq Q \times Q$. For convenience of notation, we denote $x \sim x'$ if $(x, x') \in \sim$. The subset $X \subseteq Q$ is called an *equivalent class* if $x, x' \in X \Leftrightarrow x \sim x'$. We denote by Q/\sim the set labeling all equivalent classes and define a map $\text{eq} : Q/\sim \rightarrow 2^Q$ such that $\text{eq}(X_\sim)$ is the set of states in the equivalence class $X_\sim \in Q/\sim$.

Definition 2.2. We say that a relation \sim is *observation preserving* if for any $x, x' \in Q$, $x \sim x'$ implies that $h(x) = h(x')$.

A finite *partition* P of a set \mathcal{S} is a finite collection of sets $P := \{P_i\}_{i \in I}$, such that $\cup_{i \in I} P_i = \mathcal{S}$ and $P_i \cap P_j = \emptyset$ if $i \neq j$. A finite *refinement* of P is a finite partition P' of \mathcal{S} such that for each $P_i \in P'$, there exists $P_j \in P$ such that $P_i \subseteq P_j$. Note that P is a trivial refinement of itself.

A partition naturally induces a relation, and an observation preserving relation induces a quotient TS. Given a TS $\mathcal{T} = (Q, \rightarrow, \Pi, h)$, a partition P of Q induces a relation \sim , such that $x \sim x'$ if and only if there exists $P_i \in P$ and $x, x' \in P_i$. If \sim induced by P is observational preserving, then P is said to be an observation preserving partition. One can immediately verify that a refinement of an observation preserving partition is also observation preserving.

Definition 2.3. Given a TS $\mathcal{T} = (Q, \rightarrow, \Pi, h)$ and an observation preserving relation \sim , a quotient transition system $\mathcal{T}/\sim = (Q/\sim, \rightarrow_\sim, \Pi, h_\sim)$ is a transition system, where

- Q/\sim is the set labeling all equivalent classes;
- \rightarrow_\sim is defined as follows: given $X_\sim, Y_\sim \in Q/\sim$, $X_\sim \rightarrow_\sim Y_\sim$ if and only if there exists $x \in \text{eq}(X_\sim)$ and $x' \in \text{eq}(Y_\sim)$ such that $x \rightarrow x'$;
- The set of observations Π is inherited from \mathcal{T} ;
- $h_\sim(X_\sim) := h(x)$, where $x \in \text{eq}(X_\sim)$ (note that this map is only well-defined if \sim is observation preserving).

Definition 2.4. Given a TS $\mathcal{T} = (Q, \rightarrow, \Pi, h)$, a relation \sim is a bisimulation relation of \mathcal{T} if (1) \sim is observation preserving; and (2) for any $x_1, x_2 \in Q$, if $x_1 \sim x_2$ and $x_1 \rightarrow x'_1$, then there exists $x'_2 \in Q$ such that $x_2 \rightarrow x'_2$ and $x'_1 \sim x'_2$.

If \sim is a bisimulation, then the quotient transition system \mathcal{T}/\sim is called a *bisimulation quotient* of \mathcal{T} . In this case, \mathcal{T} and \mathcal{T}/\sim are said to be *bisimilar*. Bisimulation is a very strong equivalence relation between systems. In particular, it implies language equivalence. Specifically, we have

$$\forall X_\sim \in Q/\sim, \quad \mathcal{L}_{\mathcal{T}}(\text{eq}(X_\sim)) = \mathcal{L}_{\mathcal{T}/\sim}(X_\sim). \quad (3)$$

This fact (see [Milner, 1989, Browne et al., 1988, Davoren and Nerode, 2000]) ensures that a bisimulation preserves properties expressed in temporal logics such as LTL, CTL and μ -calculus. As such, it is used as an important tool to reduce the complexity of system analysis or verification, since the bisimulation quotient (which may be finite and thus much smaller) can be analyzed or verified instead of the original system.

2.2 Polyhedral Lyapunov functions

Consider an autonomous discrete-time system,

$$x_{k+1} = \Phi(x_k), \quad k \in \mathbb{Z}_+, \quad (4)$$

where $x_k \in \mathbb{R}^n$ is the state at the discrete-time instant k and $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is an arbitrary map with $\Phi(0) = 0$. Given a state $x \in \mathbb{R}^n$, then $x' := \Phi(x)$ is called the *successor* state of x .

A function $\phi : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ belongs to class \mathcal{K}_∞ if it is continuous, strictly increasing, $\phi(0) = 0$ and $\lim_{s \rightarrow \infty} \phi(s) = \infty$.

Definition 2.5. We call a set $\mathcal{P} \subseteq \mathbb{R}^n$ *positively invariant* for system (4) if for all $x \in \mathcal{P}$ it holds that $\Phi(x) \in \mathcal{P}$. Let $\lambda \in [0, 1]$. We call $\mathcal{P} \subseteq \mathbb{R}^n$ λ -*contractive* (or shortly, *contractive*) if for all $x \in \mathcal{P}$ it holds that $\Phi(x) \in \lambda\mathcal{P}$.

The proof for the following theorem can be found in [Jiang and Wang, 2002, Lazar, 2006].

Theorem 2.1. Let \mathcal{X} be a positively invariant set for (4) with $0 \in \text{int}(\mathcal{X})$. Furthermore, let $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$, $\rho \in (0, 1)$ and $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$ such that:

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \forall x \in \mathcal{X}, \quad (5)$$

$$V(\Phi(x)) \leq \rho V(x), \forall x \in \mathcal{X}. \quad (6)$$

Then system (4) is asymptotically stable in \mathcal{X} .

Definition 2.6. A function $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$ is called a *Lyapunov function* (LF) in \mathcal{X} if it satisfies (5) and (6). If $\mathcal{X} = \mathbb{R}^n$, then V is called a *global Lyapunov function*.

The parameter ρ is called the *contraction rate* of V . For any $\Gamma > 0$, $\mathcal{P}_\Gamma := \{x \in \mathbb{R}^n \mid V(x) \leq \Gamma\}$ is called a *sublevel set* of V .

For the remainder of this paper we consider LFs defined using the infinity norm, i.e.,

$$V(x) = \|Lx\|_\infty, \quad L \in \mathbb{R}^{l \times n}, l \geq n, \quad (7)$$

where L has full-column rank. Notice that infinity norm Lyapunov functions belong to a particular class of 0-symmetric polyhedral Lyapunov functions. We opted for this type of function to simplify the exposition but in fact, the proposed abstraction method applies to general polyhedral Lyapunov functions defined by Minkowski (gauge) functions of polytopes in \mathbb{R}^n with the origin in their interior.

Proposition 2.1. Suppose that $L \in \mathbb{R}^{l \times n}$ has full-column rank and V as defined in (7) is a global LF for system (4) with contraction rate $\rho \in (0, 1)$. Then for all $\Gamma > 0$ it holds that \mathcal{P}_Γ is a polytope and $0 \in \text{int}(\mathcal{P}_\Gamma)$. Moreover, if $\Phi(x) = Ax$ for some $A \in \mathbb{R}^{n \times n}$, then for all $\Gamma > 0$ it holds that \mathcal{P}_Γ is a ρ -contractive polytope for (4).

The proof of the above result is a straightforward application of results in [Blanchini, 1994, Lazar, 2010].

3. PROBLEM FORMULATION

In this paper, we consider autonomous discrete-time linear and time-invariant (LTI) systems, i.e.,

$$x_{k+1} = Ax_k, \quad k \in \mathbb{Z}_+, \quad (8)$$

where $A \in \mathbb{R}^{n \times n}$ is a strictly stable (i.e., Schur) matrix. In this paper, we assume that a global polyhedral Lyapunov function (LF) of the form (7) with contraction rate $\rho \in$

$(0, 1)$ is known for system (8) (see Sec. 2.2). The algorithm proposed in [Lazar, 2010] is employed to construct such a function with a desired contraction rate.

Let \mathcal{X} be a polytope $\mathcal{X} := \{x \mid \|Lx\|_\infty \leq \Gamma_{\mathcal{X}}\}$ and \mathcal{D} be a polytope $\mathcal{D} := \{x \mid \|Lx\|_\infty \leq \Gamma_{\mathcal{D}}\}$, where L corresponds to the polytopic LF (7) of system (8) and we assume that $0 < \Gamma_{\mathcal{D}} < \Gamma_{\mathcal{X}}$. Note that $\mathcal{D} \subset \mathcal{X}$ and $0 \in \text{int}(\mathcal{D}) \subset \text{int}(\mathcal{X})$. We call \mathcal{X} the *working set* and \mathcal{D} the *target set*. We are interested in analyzing and verifying the behavior of the system within \mathcal{X} with respect to polytopic regions in the state space, until the target set \mathcal{D} is reached (since \mathcal{D} is positively invariant, the system trajectory will remain within \mathcal{D} after it reaches \mathcal{D}). Note that we can pick $\Gamma_{\mathcal{D}}$ arbitrarily small and $\Gamma_{\mathcal{X}}$ arbitrary large so as to capture any compact relevant subset of \mathbb{R}^n .

Remark 3.1. Our results can be extended to arbitrary positively invariant sets \mathcal{X} and \mathcal{D} , i.e., not obtained as the sublevel sets of (7). We chose to work with sublevel sets of the given polyhedral LF for the simplicity of presentation, and because such LFs allows us to easily construct a polytopic positive invariant set of any size.

We assume that there exists a set \mathcal{R} of polytopes indexed by a finite set R , i.e., $\mathcal{R} := \{\mathcal{R}_i\}_{i \in R}$, where $\mathcal{R}_i \subseteq \mathcal{X} \setminus \mathcal{D}$ for all $i \in R$, and $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ for any $i \neq j$.

Example 3.1. Consider a system as in (8) with $A = \begin{pmatrix} 0.65 & 0.32 \\ -0.42 & -0.92 \end{pmatrix}$. A polyhedral Lyapunov function was constructed with the method in [Lazar, 2010], where,

$$L = \begin{pmatrix} -0.0625 & 0.6815 & 0.9947 & 0.9947 \\ 1 & 1 & 0.6868 & -0.0678 \end{pmatrix}^T,$$

and $\rho = 0.94$. We chose $\Gamma_{\mathcal{X}} = 10$ and $\Gamma_{\mathcal{D}} = 5.063$. We show the polytope \mathcal{X} , \mathcal{D} , and a set of polytopes \mathcal{R} in Fig. 1.

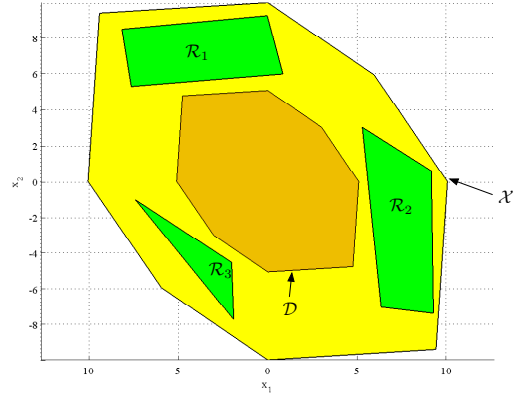


Fig. 1. An example in \mathbb{R}^2 of the working set \mathcal{X} (in yellow), the target set \mathcal{D} (in brown), and a set of observational relevant polytopes $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$ (in green).

The set \mathcal{R} represents regions of interest in the relevant state space, and the polytopes in \mathcal{R} are considered as observations of (8). Therefore, informally, a trajectory of (8) $x_0 x_1 \dots$ produces an infinite sequence of observations $o_0 o_1 \dots$, such that o_i is the index of the polytope in \mathcal{R} visited by state x_k , or $o_i = \emptyset$ if x_k is in none of the polytopes. The definition of the semantics of the system can be formalized through an embedding of (8) into a transition system, as follows.

Definition 3.1. Let \mathcal{X} , \mathcal{D} , and $\mathcal{R} = \{\mathcal{R}_i\}_{i \in R}$ be given. The embedding transition system from (8) is a transition system $\mathcal{T}_e = (Q_e, \rightarrow_e, \Pi_e, h_e)$ where

- $Q_e = \{x \in \mathbb{R}^n \mid x \in \mathcal{X}\}$
- (i) If $x \in \mathcal{X} \setminus \mathcal{D}$, then $x \rightarrow_e x'$ if and only if $x' = Ax$, i.e., x' is the state at the next time-step after applying the dynamics of (8) at x ;
- (ii) If $x \in \mathcal{D}$, $x \rightarrow_e x$ (since the target set \mathcal{D} is already reached, the behavior of the system after \mathcal{D} is reached is no longer relevant);
- $\Pi_e = R \cup \{\Pi_{\mathcal{D}}\}$, i.e., the set of observations is the set of labels of regions, plus the label $\Pi_{\mathcal{D}}$ for \mathcal{D} ;
- (i) $h_e(x) := i$ if and only if $x \in \mathcal{R}_i$;
- (ii) $h_e(x) := \emptyset$ if and only if $x \in \mathcal{X} \setminus (\mathcal{D} \cup \bigcup_{i \in R} \mathcal{R}_i)$;
- (iii) $h_e(x) := \Pi_{\mathcal{D}}$ if and only if $x \in \mathcal{D}$.

Note that \mathcal{T}_e is infinite and deterministic. Moreover, \mathcal{T}_e exactly captures the system dynamics under (8) in the relevant state space $\mathcal{X} \setminus \mathcal{D}$, since a transition of the embedding TS \mathcal{T}_e naturally corresponds to the evolution of the discrete-time system in one time-step (until the target set is reached). Indeed, the trajectory of \mathcal{T}_e from a state $x \in \mathcal{X} \setminus \mathcal{D}$ is exactly the same as the trajectory of the system from x evolved under (8) until \mathcal{D} is reached.

The state space of \mathcal{T}_e (which is the working set \mathcal{X}) can be naturally partitioned as

$$P_{\mathcal{X}} := \left\{ \left\{ \mathcal{R}_i \right\}_{i \in R}, \mathcal{X} \setminus (\mathcal{D} \cup \bigcup_{i \in R} \mathcal{R}_i), \mathcal{D} \right\}. \quad (9)$$

It is straightforward to establish from the definition of h_e in \mathcal{T}_e , that the relation induced from $P_{\mathcal{X}}$ (see Sec. 2.1) is observation preserving. We now formulate the main problem addressed in this paper.

Problem 3.1. Let a system (8) with a polyhedral Lyapunov function of the form (7), sets \mathcal{X} , \mathcal{D} and $\{\mathcal{R}_i\}_{i \in R}$ be given. Find a finite observation preserving partition P such that its induced relation \sim is a bisimulation of the embedding transition system \mathcal{T}_e , and obtain the corresponding bisimulation quotient \mathcal{T}_e/\sim .

Remark 3.2. In fact, $P_{\mathcal{X}}$ is the coarsest observation preserving partition for \mathcal{T}_e , and its induced relation is called an *observation equivalence* relation in literature. As a result, a finite partition is observation preserving if and only if it is a refinement of $P_{\mathcal{X}}$. Therefore, any solution of Prob. 3.1 is a refinement of $P_{\mathcal{X}}$.

4. GENERATING THE BISIMULATION QUOTIENT

Starting from a polyhedral Lyapunov function $V(x) = \|Lx\|_{\infty}$ with a contraction rate $\rho = (0, 1)$ as described in Sec. 2.2 for system (8), we first generate a sequence of polytopic sublevel sets of the form $\mathcal{P}_{\Gamma} := \{x \in \mathbb{R}^n \mid \|Lx\|_{\infty} \leq \Gamma\}$ as follows. Recall that $\mathcal{X} = \mathcal{P}_{\Gamma_{\mathcal{X}}}$ and $\mathcal{D} = \mathcal{P}_{\Gamma_{\mathcal{D}}}$ for some $0 < \Gamma_{\mathcal{D}} < \Gamma_{\mathcal{X}}$. We define a finite sequence $\bar{\Gamma} := \Gamma_0, \dots, \Gamma_N$, where

$$\Gamma_{i+1} = \rho^{-1} \Gamma_i, \quad i = 0, \dots, N-2, \quad (10)$$

where $\Gamma_0 := \Gamma_{\mathcal{D}}$, $\Gamma_N := \Gamma_{\mathcal{X}}$, and $N := \arg \min_N \{\rho^{-N} \Gamma_0 \mid \rho^{-N} \Gamma_0 \geq \Gamma_{\mathcal{X}}\}$. The sequence $\bar{\Gamma}$ generates a sequence of sublevel sets $\bar{\mathcal{P}}_{\Gamma} := \mathcal{P}_{\Gamma_0}, \dots, \mathcal{P}_{\Gamma_N}$. From the definition of the sublevel sets and $\bar{\Gamma}$, we have that

$$\mathcal{P}_{\Gamma_0} \subset \dots \subset \mathcal{P}_{\Gamma_N}. \quad (11)$$

Note that \mathcal{P}_{Γ_0} is exactly \mathcal{D} , \mathcal{P}_{Γ_N} is exactly \mathcal{X} , and $\mathcal{P}_{\Gamma_{N-1}}$ is the largest sublevel set defined via (10) that is a subset of \mathcal{X} .

Next, we define a *slice* of the state space as follows:

$$\mathcal{S}_i := \mathcal{P}_{\Gamma_i} \setminus \mathcal{P}_{\Gamma_{i-1}}, \quad i = 1, \dots, N-1. \quad (12)$$

For convenience, we also denote $\mathcal{S}_0 := \mathcal{P}_{\Gamma_0}$ (although \mathcal{S}_0 is not a slice in between two sublevel sets). We immediately see that the sets $\{\mathcal{S}_i\}_{i=0, \dots, N}$ form a partition of \mathcal{X} . Note that the slices are bounded semi-linear sets (see Sec. 2).

Example 4.1. (Example 3.1 continued). Consider the system and sets as given in Example 3.1. The polytopic sublevel sets $\bar{\mathcal{P}}_{\Gamma} := \mathcal{P}_{\Gamma_0}, \dots, \mathcal{P}_{\Gamma_N}$ are shown in Fig. 2.

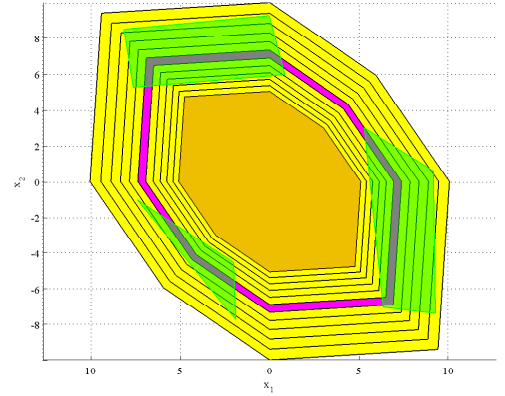


Fig. 2. An example of sublevel sets with $N = 11$ and one slice \mathcal{S}_6 (in purple).

The sublevel sets and the slices are specifically constructed as in (10) with the contractive parameter ρ , in order to provide the useful property that states within a slice must transition to a lower slice.

Proposition 4.1. Assume that the set of slices $\{\mathcal{S}_i\}_{i=0, \dots, N}$ is obtained by a sequence $\bar{\Gamma}$ satisfying (10). Given a state x in the i -th slice, i.e., $x \in \mathcal{S}_i$, where $1 \leq i \leq N$, its successor state ($x' = Ax$) satisfies $x' \in \mathcal{S}_j$ for some $j < i$.

Proof. From Prop. 2.1, we have that \mathcal{P}_{Γ_i} are ρ -contractive. By the definition of a ρ -contractive set (Def. 2.5), we have that $x' = Ax \in \rho \mathcal{P}_i = \{x \in \mathbb{R}^n \mid \|Lx\|_{\infty} \leq \rho \Gamma_i\}$. From (10), we have $\rho \Gamma_i = \Gamma_{i-1}$. Therefore $\mathcal{P}_{\Gamma_{i-1}} = \{x \in \mathbb{R}^n \mid \|Lx\|_{\infty} \leq \Gamma_{i-1}\}$ implies that $\mathcal{P}_{\Gamma_{i-1}} = \{x \in \mathbb{R}^n \mid \|Lx\|_{\infty} \leq \rho \Gamma_i\}$ and hence $\mathcal{P}_{\Gamma_{i-1}} = \rho \mathcal{P}_{\Gamma_i}$ and $x' \in \mathcal{P}_{\Gamma_{i-1}}$. From the definition of slices (12), $x' \in \mathcal{S}_j$ for some $j < i$. ■

We now present the abstraction algorithm (see Alg. 1) that computes the bisimulation quotient. In Alg. 1, we make use of two procedures **FindPre** and **Refine**, which will be further explained below. The main idea is to start with $P_{\mathcal{X}}$, and iteratively refine the partition until it becomes a refinement to both $P_{\mathcal{X}}$ as in (9) and $\{\mathcal{S}_i\}_{i=0, \dots, N}$. The first procedure is necessary so that the partition is observation preserving. The second procedure allows us to ensure that at iteration i of the algorithm, the bisimulation quotient for states within \mathcal{P}_{Γ_i} is completed. Similar to the slices, the solution to Prob. 3.1 obtained from Alg. 1 is a partition consisting of bounded semi-linear sets.

Algorithm 1 Abstraction algorithm

Input: System dynamics (8), polytopic LF $V(x) = \|Lx\|_\infty$ with a contractive rate ρ , sets \mathcal{X} , \mathcal{D} and $\{\mathcal{R}_i\}_{i \in R}$.

Output: \mathcal{T}_e/\sim as a bisimulation quotient of the embedding transition system \mathcal{T}_e and the corresponding observation preserving partition P .

- 1: Generate the sequence of sublevel sets $\tilde{P}_\Gamma = \mathcal{P}_{\Gamma_0}, \dots, \mathcal{P}_{\Gamma_N}$ and slices $\mathcal{S}_0, \dots, \mathcal{S}_N$ as defined in (12).
 - 2: Obtain $P_\mathcal{X}$ as in (9).
 - 3: Set $P_0 := \text{Refine}(P_\mathcal{X}, \{\mathcal{S}_i\}_{i=0, \dots, N})$.
 - 4: Initialize \mathcal{T}_e/\sim_0 by setting Q_e/\sim_0 as the set labeling P_0 . Set transition only for the state $q \in Q_e/\sim_0$ where $\text{eq}(q) = \mathcal{S}_0 = \mathcal{D}$ with $q \rightarrow_{\sim_0} q$.
 - 5: **for** each $i = 0, \dots, N-1$ **do**
 - 6: **for** each $\tilde{P} \in P_i$ where $\tilde{P} \subseteq \mathcal{S}_i$ **do**
 - 7: Find $P_{\text{Pre}} = \text{FindPre}(\tilde{P})$
 - 8: Set $P_{i+1} = \text{Refine}(P_i, P_{\text{Pre}})$. Update and add the corresponding states in \mathcal{T}_e/\sim_{i+1} . Set the transitions of the added states to \tilde{P} in \mathcal{T}_e/\sim_{i+1} .
 - 9: **end for**
 - 10: **end for**
 - 11: Return \mathcal{T}_e/\sim_N and P_N as a solution to Prob. 3.1.
-

Procedure **FindPre**(\tilde{P}) takes as input \tilde{P} , a bounded semi-linear set (e.g., a slice), and returns the set $\text{Pre}_{\mathcal{T}_e}(\tilde{P})$. In general, the Pre of a semi-linear set is a semi-linear set, and it can be computed via quantifier elimination [Bochnak et al., 1998]. In particular, a bounded semi-linear set \tilde{P} implies that it only belongs to one of the following cases:

- (i) If \tilde{P} is a polytope \mathcal{P} in the representation $\mathcal{P} = \{x \in \mathbb{R}^n \mid H_\mathcal{P}x \leq h_\mathcal{P}\}$ for some $k \geq n+1$, $H_\mathcal{P} \in \mathbb{R}^{k \times n}$ and $h_\mathcal{P} \in \mathbb{R}^k$, the Pre of \mathcal{P} can be obtained using polytopic operations only, as
$$\text{Pre}_{\mathcal{T}_e}(\mathcal{P}) = \{x \in \mathbb{R}^n \mid H_\mathcal{P}Ax \leq h_\mathcal{P}\}, \quad (13)$$
which is a possibly degenerate polytope in \mathbb{R}^n . Note that (13) applies to a polytope \mathcal{P} of any dimension;
- (ii) If \tilde{P} is a union of polytopes, one can use a standard convexation method to decompose \tilde{P} to a set of polytopes $\{\mathcal{P}_i\}_{i \in I}$ (see, e.g., [Grünbaum, 2003]). The Pre of \tilde{P} can then be computed as $\cup_{i \in I} \text{Pre}_{\mathcal{T}_e}(\mathcal{P}_i)$ using (13);
- (iii) If \tilde{P} is a convex and bounded semi-linear set, then $\tilde{P} = \mathcal{P} \setminus \cup_{i \in I} \mathcal{P}_i$ for some polytope \mathcal{P} and its facet $\mathcal{P}_i \in f(\mathcal{P})$. Since \mathcal{T}_e is deterministic, we have $\text{Pre}_{\mathcal{T}_e}(\tilde{P}) = \text{Pre}(\mathcal{P}) \setminus \text{Pre}(\cup_{i \in I} \mathcal{P}_i)$, where the second term can be computed as described in case (ii);
- (iv) If \tilde{P} is a general (non-convex) bounded semi-linear set, then again it can be decomposed into convex and bounded semi-linear sets and $\text{Pre}_{\mathcal{T}_e}(\tilde{P})$ can be computed as the union of their Pres as described in case (iii).

As summarized above, we see that **FindPre**(\tilde{P}) can always be carried out by convex decompositions and repeated applications of (13), and thus **FindPre**(\tilde{P}) only requires polytopic operations. Since the Pre of a bounded semi-linear set is a bounded semi-linear set, **FindPre** can be carried out with polytopic operations throughout Alg. 1.

The procedure **Refine**(P, \tilde{P}) (outlined in Alg. 2) refines an observation preserving partition P by partitioning the set \tilde{P} , which is assumed to be a bounded semi-linear set¹. The proof of correctness of Alg. 2 is straight-forward, since sets in a partition $P = \{P_i\}_{i \in I}$ are piecewise disjoint by definition and as such $\tilde{P} = \bigcup_{i \in I} (P_i \cap \tilde{P})$. If P consists of bounded semi-linear sets, we can directly see from Alg. 2 that the resultant refinement P' has the same property. This fact allows us to use **FindPre**(\tilde{P}) for each set $\tilde{P} \in P'$.

Algorithm 2 $P' = \text{Refine}(P, \tilde{P})$

Input: P is an observation preserving partition of \mathcal{X} . $\tilde{P} \subseteq \mathcal{X}$ is a bounded semi-linear set.

Output: $P' = \{P'_i\}_{i \in I}$ is a finite refinement of P , and there exists $J \subseteq I$ such that $\tilde{P} = \cup_{j \in J} P'_j$.

- 1: Set $P' = P$
 - 2: **for all** $P'_i \in P'$ such that $P'_i \cap \tilde{P} \neq \emptyset$ **do**
 - 3: Replace P'_i in P' by $\{P'_i \cap \tilde{P}, P'_i \setminus \tilde{P}\}$
 - 4: **end for**
-

The correctness of Alg. 1 will be shown by an inductive argument. Given a sublevel set \mathcal{P}_{Γ_i} and a partition P_i as obtained in Alg. 1, we define \tilde{P}_i as

$$\tilde{P}_i := \{P \in P_i \mid P \subseteq \mathcal{P}_{\Gamma_i}\}. \quad (14)$$

From Alg. 1, we see that P_0 partitions all the slices, and since P_i is a finite refinement of P_0 , we can directly see that \tilde{P}_i is a partition of \mathcal{P}_{Γ_i} . We define an embedding transition system $\mathcal{T}_e(i)$ as a subset of \mathcal{T}_e , where its state-space is $\{x \in Q_e \mid x \in \mathcal{P}_{\Gamma_i}\}$. We have the following proposition.

Proposition 4.2. At the completion of the i -th iteration (in the outer loop) of Alg. 1 (where P_{i+1} is obtained), if \sim_i induced by \tilde{P}_i as defined in (14) is a bisimulation of $\mathcal{T}_e(i)$, then \sim_{i+1} induced by \tilde{P}_{i+1} is a bisimulation of $\mathcal{T}_e(i+1)$.

Proof. If \sim_i induced by P_i is a bisimulation of $\mathcal{T}_e(i)$, then from Prop. 4.1, we have that for each $x \in \mathcal{S}_{i+1}$, $x' = Ax$ must be in a lower slice and thus $x' \in \mathcal{T}_e(i)$. For each $x' = Ax$ where $x \in \mathcal{S}_{i+1}$, if $x' \in \mathcal{S}_i$, then we have $x \in P_{\text{Pre}} = \text{FindPre}(P)$ (from Step 7 of Alg. 1) for some $P \in P_i$, and after the refinement step (Step 8), we have $x \in P' \subseteq P_{\text{Pre}}$ for some $P' \in P_{i+1}$, and \mathcal{T}_e/\sim_{i+1} is updated by 1) adding state $\text{eq}(P')$ to Q_e/\sim_{i+1} and 2) adding the transition $\text{eq}(P') \rightarrow_{\sim_{i+1}} \text{eq}(P)$. We note that from the definition of Pre, for any $x \in P'$, $x' = Ax \in P$, thus for any $x_i \sim x_j$, $Ax_i \sim Ax_j$, and transition $\text{eq}(P') \rightarrow_{\sim_{i+1}} \text{eq}(P)$ satisfies the bisimulation requirement. On the other hand, if $x' \notin \mathcal{S}_i$, then $x' \in \mathcal{S}_j$ for some $j < i$ and x is already in a set P' where $\text{eq}(P') \rightarrow_{\sim_{i+1}} \text{eq}(P)$ for some P satisfying the bisimulation requirement. Therefore, step 7 and 8 of Alg. 1 provides exactly the transitions needed for states all states in \mathcal{S}_{i+1} and thus, \sim_{i+1} induced by \tilde{P}_{i+1} is a bisimulation of $\mathcal{T}_e(i+1)$.

Proposition 4.3. Alg. 1 returns a solution to Prob. 3.1 in finite time.

¹ With a slight abuse of notation, **Refine**($P, \{\tilde{P}\}_{i \in I}$) stands for sequentially applying **Refine**(P, \tilde{P}_i) for each $i \in I$.

Proof. From Alg. 2, we have that P_i is a refinement of $P_{\mathcal{X}}$ for any $i = 0, \dots, N$. Therefore, P_N and its induced relation \sim_N are observational preserving.

At step 4 of Alg. 1, we set $q \rightarrow_{\sim_0} q$ where $\text{eq}(q) = \mathcal{D}$. From the definition of \mathcal{T}_e , we see that since \mathcal{D} is the only state, \sim_0 induced by \tilde{P}_0 is a bisimulation of $\mathcal{T}_e(0)$. Using Prop. 4.2 and induction, at iteration $N - 1$, we have that \sim_N induced by \tilde{P}_N is a bisimulation of $\mathcal{T}_e(N)$. Note that \tilde{P}_N is exactly P_N , \mathcal{P}_{Γ_N} is exactly \mathcal{X} and $\mathcal{T}_e(N)$ is exactly \mathcal{T}_e . Therefore \sim_N induced by P_N is a bisimulation of \mathcal{T}_e .

Finally, note that at each iteration, the number of sets updated are finite. Therefore, the bisimulation quotient is finite and moreover Alg. 1 completes in finite time. ■

Example 4.2. (Example 4.1 continued). Alg. 1 is applied on the same setting as in Example 4.1 to compute the bisimulation quotient. “Snapshots” of the algorithm iterations are shown in Fig. 3. The final result is a Transition system with 320 states. In this example, Alg. 1 was completed in 3 minutes on a Macbook Pro 2011 model.

5. SYSTEM VERIFICATION WITH LINEAR TEMPORAL LOGIC FORMULAS

In this section we show how we can use the bisimulation quotient obtained as a solution to Prob. 3.1 to verify the behavior of system (8) in the state space $\mathcal{X} \setminus \mathcal{D}$ over the observed regions $\{\mathcal{R}_i\}_{i \in R}$ and the observation $\Pi_{\mathcal{D}}$ corresponding to \mathcal{D} . We will employ Linear Temporal Logic (LTL) to describe high level system specifications. A detailed description of the syntax and semantics of LTL is beyond the scope of this paper and can be found in, for example, [Clarke et al., 1999]. Roughly, an LTL formula is built up from a set of atomic propositions Π , which are properties that can be either true or false, standard Boolean operators \neg (negation), \vee (disjunction), \wedge (conjunction), and temporal operators \mathbf{X} (next), \mathbf{U} (until), \mathbf{F} (eventually), \mathbf{G} (always) and \Rightarrow (implication). The semantics of LTL formulas are given over words, which is defined as an infinite sequence $\mathbf{o} = o_0 o_1 \dots$, where $o_i \in 2^\Pi$ for all i . We say $\mathbf{o} \models \phi$ if the word \mathbf{o} satisfies the LTL formula ϕ . We say a trajectory \mathbf{q} of a transition system \mathcal{T} satisfies LTL formula ϕ , if the word generated by \mathcal{T} (see Def. 2.1) satisfies ϕ .

Example 5.1. Again, consider the setting in Example 3.1 with $\mathcal{R} = \{\mathcal{R}_i\}_{i=\{1,2,3\}}$. We now consider a specification in LTL over $R = \{1, 2\}$. For example, the specification:

“The system trajectory never visits Region 2 and eventually visits Region 1. Moreover, if it visits Region 3 then it must not visit Region 1 at the next consecutive time instant”

can be translated to an LTL formula:

$$\phi := \mathbf{G} \neg 2 \wedge \mathbf{F} 1 \wedge (3 \Rightarrow \mathbf{X} \neg 1) \quad (15)$$

Remark 5.1. Set \mathcal{D} is by definition positively invariant. Therefore, all trajectories of (8) eventually reach \mathcal{D} . As a result, we see that any LTL formula satisfiable by (8) must not violate formula $\mathbf{F} \Pi_{\mathcal{D}}$. For example $\psi = \mathbf{G} \neg \Pi_{\mathcal{D}} \wedge \phi$ is not satisfiable by the system for any LTL formula ϕ as the first part of ψ is in contradiction to $\mathbf{F} \Pi_{\mathcal{D}}$.

Problem 5.1. Let system (8) with a polyhedral Lyapunov function in the form of (7), sets \mathcal{X} , \mathcal{D} and $\{\mathcal{R}_i\}_{i \in R}$,

and an LTL formula ϕ over $R \cup \Pi_{\mathcal{D}}$ be given. Find the largest set $\mathcal{S} \subseteq Q_e$ such that state trajectories of the embedding transition system \mathcal{T}_e originating from \mathcal{S} satisfy ϕ .

Our solution to Prob. 5.1 proceed by finding a bisimulation quotient \mathcal{T}/\sim of the embedding transition system \mathcal{T}_e using Alg. 1. Then we translate ϕ to a so-called Büchi Automaton, defined below.

Definition 5.1. A (non-deterministic) Büchi automaton is a tuple $\mathcal{B} = (S_{\mathcal{B}}, S_{\mathcal{B}0}, \Sigma, \delta, F_{\mathcal{B}})$, where

- $S_{\mathcal{B}}$ is a finite set of states;
- $S_{\mathcal{B}0} \subseteq S_{\mathcal{B}}$ is the set of initial states;
- Σ is the input alphabet;
- $\delta : S_{\mathcal{B}} \times \Sigma \rightarrow 2^{S_{\mathcal{B}}}$ is the transition function;
- $F_{\mathcal{B}} \subseteq S_{\mathcal{B}}$ is the set of accepting states.

We denote $s \xrightarrow{\sigma}_{\mathcal{B}} s'$ if $s' \in \delta(s, \sigma)$. A word $\sigma_0 \sigma_1 \dots$ over Σ generates trajectories $s_0 s_1 \dots$ where $s_0 \in S_{\mathcal{B}0}$ and $s_k \xrightarrow{\sigma_k}_{\mathcal{B}} s_{k+1}$ for all $k \geq 0$. \mathcal{B} accepts a word over Σ if it generates at least one trajectory on \mathcal{B} that intersects $F_{\mathcal{B}}$ infinitely many times.

For any LTL formula ϕ over Π , one can construct a Büchi automaton with input alphabet $\Sigma = 2^\Pi$ accepting all and only words over 2^Π satisfying ϕ [Clarke et al., 1999]. Algorithms and implementations for the translation from ϕ to a corresponding Büchi automaton \mathcal{B} can be found in [Gastin and Oddoux, 2001].

Definition 5.2. Given a transition system $\mathcal{T} = (Q, \rightarrow, \Pi, h)$ and a Büchi automaton $\mathcal{B} = (S_{\mathcal{B}}, S_{\mathcal{B}0}, 2^\Pi, \delta_{\mathcal{B}}, F_{\mathcal{B}})$, their product automaton, denoted by $\mathcal{A} = \mathcal{T} \times \mathcal{B}$, is a tuple $\mathcal{A} = (S_{\mathcal{A}}, S_{\mathcal{A}0}, \Delta_{\mathcal{A}}, F_{\mathcal{A}})$ where

- $S_{\mathcal{A}} = Q \times S_{\mathcal{B}}$;
- $S_{\mathcal{A}0} = Q \times S_{\mathcal{B}0}$;
- $\Delta_{\mathcal{A}} \subseteq S_{\mathcal{A}} \times S_{\mathcal{A}}$ is the set of transitions, defined by:
 $((q, s), (q', s')) \in \Delta_{\mathcal{A}}$ iff $q \rightarrow q'$ and $s \xrightarrow{h(q)}_{\mathcal{B}} s'$;
- $F_{\mathcal{A}} = Q \times F_{\mathcal{B}}$.

We denote $(q, s) \rightarrow_{\mathcal{A}} (q', s')$ if $((q, s), (q', s')) \in \Delta_{\mathcal{A}}$. A trajectory $\mathbf{p} = (q_0, s_0)(q_1, s_1) \dots$ of \mathcal{A} is an infinite sequence such that $(q_0, s_0) \in S_{\mathcal{A}0}$ and $(q_k, s_k) \rightarrow_{\mathcal{A}} (q_{k+1}, s_{k+1})$ for all $k \geq 0$. Trajectory \mathbf{p} is called accepting if and only if it intersects $F_{\mathcal{A}}$ infinitely many times.

By the construction of \mathcal{A} from \mathcal{T} and \mathcal{B} , \mathbf{p} is accepted if and only if $\mathbf{q} = \gamma_{\mathcal{T}}(\mathbf{p})$ satisfies the LTL formula corresponding to \mathcal{B} [Clarke et al., 1999], where $\gamma_{\mathcal{T}}(\mathbf{p})$ is the projection of a trajectory \mathbf{p} on \mathcal{P} onto \mathcal{T} by simply removing the automaton part of the state in $(q, s) \in S_{\mathcal{A}}$.

Remark 5.2. Normally the product automaton is constructed from a transition system with an initial state q_0 , whereas the transition system generated as a solution to Prob. 3.1 is not initialized. Since any state $q \in Q_e/\sim$ can be an initial condition, the set of initial states of \mathcal{A} is $Q_e/\sim \times S_{\mathcal{B}0}$. Thus, here we augment the definition of \mathcal{A} slightly so that it is constructed as a product of an uninitialized transition system and a Büchi automaton.

In [Ding et al., 2010], an algorithm was proposed to compute the largest subset $F_{\mathcal{A}}^* \subseteq F_{\mathcal{A}}$ such that it can reach another state in $F_{\mathcal{A}}^*$. The following property was shown to hold:

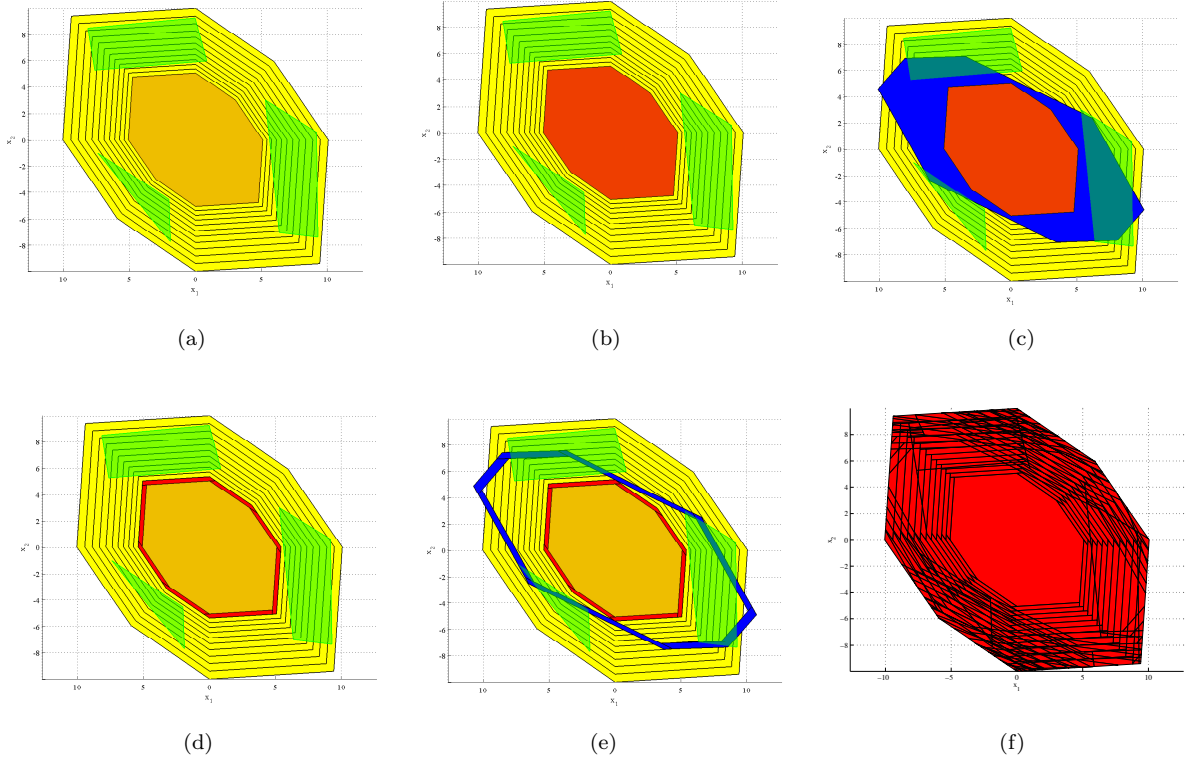


Fig. 3. (a) The slices are shown in yellow, except \mathcal{D} , which is shown in light brown. The observed regions are shown in transparent green. (b) In the first iteration ($i = 0$), the slice $\mathcal{S}_0 = \mathcal{D}$ is shown in red. (c) The Pre of \mathcal{S}_0 is shown in blue. At this point, the bisimulation quotient for states within $P_{\Gamma_0} = \mathcal{D}$ is completed, which consists of just a single state. (d) In the second iteration ($i = 1$), the slice \mathcal{S}_1 is shown in red. (e) The Pre of \mathcal{S}_1 is shown in blue. At this point, the bisimulation quotient for states within P_{Γ_1} is completed. (f) At the last iteration where $i = 10$, the algorithm is completed. The state space covered by the bisimulation quotient is shown in red, covering all of \mathcal{X} .

Proposition 5.1. A trajectory \mathbf{p} is accepting if and only if each accepting state appearing in \mathbf{p} is in $F_{\mathcal{A}}^*$.

A state $q \in Q$ of \mathcal{T} from which the trajectory satisfies the formula must be such that a state in $F_{\mathcal{A}}^*$ is reachable from (q, s_0) for some $s_0 \in S_{B0}$. Therefore, Prob. 5.1 can be solved by a simple reachability analysis for the set $F_{\mathcal{A}}^*$ on the product automaton. Note that during the generation of set $F_{\mathcal{A}}^*$ in the algorithm proposed in [Ding et al., 2010], the reachability is already determined for each state in \mathcal{A} , so no extra computation is necessary. This procedure is summarized in the following algorithm.

Algorithm 3 Finding the largest subset satisfying an LTL formula

Input: \mathcal{X} , \mathcal{D} , $\{\mathcal{R}_i\}_{i \in R}$, and an LTL formula ϕ over $R \cup \Pi_{\mathcal{D}}$

Output: The largest set $\mathcal{S} \subseteq Q_e$, such that the embedding transition system \mathcal{T}_e with the initial state $q_0 \in \mathcal{S}$ produces a word satisfying ϕ

- 1: Generate the bisimulation quotient \mathcal{T}_e/\sim for \mathcal{T}_e .
- 2: Translate ϕ to a Buchi automaton \mathcal{B}
- 3: Generate the product \mathcal{A} between \mathcal{T}_e/\sim and \mathcal{B}
- 4: Find the subset $F_{\mathcal{A}}^* \subseteq F_{\mathcal{A}}$ with the algorithm by Ding et al. [2010].
- 5: $\mathcal{S} = \{eq(q) \mid q \in \mathcal{T}_e/\sim \text{ and there exists } s_0 \in S_{B0} \text{ such that } F_{\mathcal{A}}^*(q, s_0) \text{ is reachable from } (q, s_0)\}$

Proposition 5.2. Upon termination, Alg. 3 gives a solution to Prob. 5.1.

Proof. We prove that Alg. 3 generates the largest set of satisfying states by contradiction. From the last step of Alg. 3, we have that $\mathcal{S} = \{eq(q) \mid q \in \mathcal{T}_e/\sim \text{ and } \exists s_0 \in S_{B0} \text{ such that } F_{\mathcal{A}}^*(q, s_0) \text{ is reachable from } (q, s_0)\}$. Assume that there exists $q_e \notin \mathcal{S}$ such that a trajectory from q_e satisfies ϕ , and $q_e \in eq(q)$ where $q \in Q_e/\sim$. In this case, on the product $\mathcal{T}_e/\sim \times \mathcal{B}$, from a state $(q, s_0) \in S_{A0}$, a state in $F_{\mathcal{A}}^*$ cannot be reached, and from Prop. 5.1, we have that trajectory \mathbf{p} cannot be accepting on $\mathcal{T}_e/\sim \times \mathcal{B}$ and $\gamma_{\mathcal{T}_e/\sim}(\mathbf{p})$ as a trajectory of \mathcal{T}_e/\sim cannot be accepting. Therefore, $\mathcal{L}_{\mathcal{T}_e/\sim}(q)$ does not satisfy ϕ . By the property of language equivalence of bisimulations, we have $\mathcal{L}_{\mathcal{T}_e}(q_e) \subseteq \mathcal{L}_{\mathcal{T}_e}(eq(q)) = \mathcal{L}_{\mathcal{T}_e/\sim}(q)$, and therefore the trajectory from q_e cannot be accepting, which violates the above assumption. ■

Example 5.2. (Example 5.1 continued). For the example specification ϕ as in (15), we obtained the solution to Prob. 5.1 by following Alg. 3. The set of initial states from which the state trajectories satisfy (15) are shown in Fig. 4.

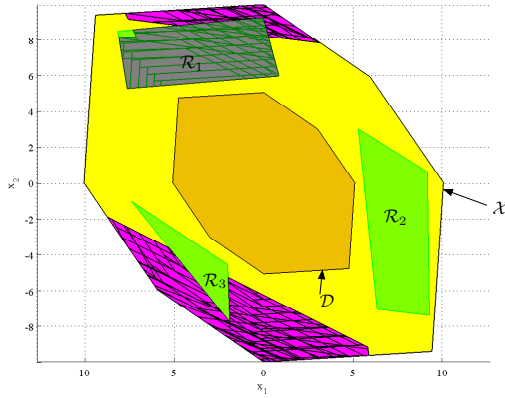


Fig. 4. The set of states satisfying ϕ (in purple).

6. CONCLUSIONS AND FINAL REMARKS

In this paper we presented a method to abstract the behavior of an autonomous linear system within a positively invariant subset of \mathbb{R}^n to a finite transition system via bisimulation. We employed polyhedral Lyapunov functions to guide the partitioning of the state space and showed that this results requires only polytopic operations.

Future work deals with an extension to continuous-time linear systems and other classes of systems that admit polyhedral Lyapunov functions, in particular, switched linear systems. We also aim to relax some assumptions and improve the computational complexity of the approach by reducing the size of the bisimulation quotient.

REFERENCES

- R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88:971–984, 2000.
- G. Batt, D. Ropers, H. De Jong, J. Geiselmann, R. Ma-teescu, M. Page, and D. Schneider. Validation of qualitative models of genetic regulatory networks by model checking: Analysis of the nutritional stress response in escherichia coli. *Bioinformatics*, 21(suppl 1):i19–i28, 2005.
- C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G.J. Pappas. Symbolic planning and control of robot motion [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):61–70, 2007. ISSN 1070-9932.
- F. Blanchini. Ultimate boundedness control for uncertain discrete-time systems via set-induced Lyapunov functions. *IEEE Transactions on Automatic Control*, 39(2): 428–433, 1994.
- J. Bochnak, M. Coste, and M.F. Roy. *Real algebraic geometry*, volume 36. Springer Verlag, 1998.
- M.C. Browne, E.M. Clarke, and O. Grumberg. Characterizing finite kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59(1-2):115–131, 1988.
- E. M. Clarke, D. Peled, and O. Grumberg. *Model checking*. MIT Press, 1999.
- J.M. Davoren and A. Nerode. Logics for hybrid systems. *Proceedings of the IEEE*, 88(7):985–1010, 2000.
- Xu Chu Ding, Calin Belta, and Christos G. Cassandras. Receding horizon surveillance with temporal logic specifications. In *IEEE Conference on Decision and Control*, pages 256–261, December 2010.
- P. Gastin and D. Oddoux. Fast LTL to Buchi automata translation. *Lecture Notes in Computer Science*, pages 53–65, 2001.
- B. Grünbaum. *Convex polytopes*, volume 221. Springer Verlag, 2003.
- Z.P. Jiang and Y. Wang. A converse lyapunov theorem for discrete-time systems with disturbances. *Systems & control letters*, 45(1):49–58, 2002.
- M. Lazar. *Model predictive control of hybrid systems: Stability and robustness*. PhD thesis, Eindhoven University of Technology, 2006.
- M. Lazar. On infinity norms as lyapunov functions: Alternative necessary and sufficient conditions. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5936–5942. IEEE, 2010.
- S. G. Loizou and K. J. Kyriakopoulos. Automatic synthesis of multiagent motion tasks based on LTL specifications. In *IEEE Conference on Decision and Control*, pages 153–158, Paradise Islands, The Bahamas, December 2004.
- R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
- G. J. Pappas. Bisimilar linear systems. *Automatica*, 39(12):2035–2047, 2003.
- C. Sloth and R. Wisniewski. Abstraction of continuous dynamical systems utilizing lyapunov functions. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 3760–3765. IEEE, 2010.
- P. Tabuada and G.J. Pappas. Linear time logic control of discrete-time linear systems. *Automatic Control, IEEE Transactions on*, 51(12):1862–1877, 2006.
- G.M. Ziegler. *Lectures on polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995.